Pattern Analysis and Applications (2021) 24:1757–1767
https://doi.org/10.1007/s10044-021-01004-9

SHORT PAPER

ChoiceNet: CNN learning through choice of multiple feature map representations

Farshid Rayhan[1] · **Aphrodite Galata**[1] · **Tim F. Cootes**[2]

Received: 12 October 2020 / Accepted: 3 June 2021 / Published online: 11 July 2021

Abstract
We introduce a new architecture called ChoiceNet where each layer of the network is highly connected with skip connections and channelwise concatenations. This enables the network to alleviate the problem of vanishing gradients, reduces the number of parameters without sacrificing performance and encourages feature reuse. We evaluate our proposed architecture on three independent tasks: classification, segmentation and facial landmark localisation. For this, we use benchmark datasets such as ImageNet, CIFAR-10, CIFAR-100, SVHN CamVid and 300W.

Keywords Classification · Segmentation · Network architecture

1 Introduction

Convolutional networks have become a dominant approach for visual object recognition [16, 30, 50, 52]. However, as convolutional neural networks (CNNs) are becoming increasingly deep, the vanishing gradient problem [16] poses significant challenges as input information can vanish passing through many layers before reaching the end.

When training a deep neural network, gradients can become very small during the backpropagation process, making it hard to optimise the parameters in the early stages of the network. Therefore, in the training phase the weights of the layers at the end of the network get updated quite rapidly while the early layers do not, leading to poor results. Activation function 'ReLU' and regularisation methods like dropouts were proposed to address this problem [11]. However, while these methods are important they do not solve the problem entirely. Huang et al. [19] found that as layers are added to a network, at some point its performance will start to decrease [19]. Recent work [16, 18, 53, 54] proposed different solutions such as skip connections [16], use of different sized filters in parallel [53, 54] and exhaustive concatenation between layers [18]. This goes some way to addressing the problem.

In this paper, we draw inspiration from the above networks [16–18, 53–55] and propose a novel network architecture that retains positive aspects of these approaches [16, 18] while overcoming some of their limitations. Figure 1 illustrates a single module layout of our proposed architecture where its unique connectivity is displayed. We show that ChoiceNet design allows good gradient and information flow through the network while using fewer parameters compared to other state-of-the-art schemes. We evaluate ChoiceNet on benchmark datasets (ImageNet [28], CIFAR10 [27], CIFAR 100 [27] and SVHN [40]) for image classification, 300W [47] for facial landmark localisation and CamVid dataset [22] for semantic segmentation. Our model performs well against existing networks [14, 16, 18, 53–55] on all these datasets, showing promising results when compared to the current state-of-the-art (Fig. 2).

This document is the results of the research project funded by Dept. of Computer Science, The University of Manchester and Toyota Motor Europe.

✉ Farshid Rayhan
f.rayhan@manchester.ac.uk

Aphrodite Galata
A.Galata@manchester.ac.uk

Tim F. Cootes
timothy.f.cootes@manchester.ac.uk

[1] Department of Computer Science, The University of Manchester, Manchester, UK

[2] School of Health Sciences, The University of Manchester, Manchester, UK

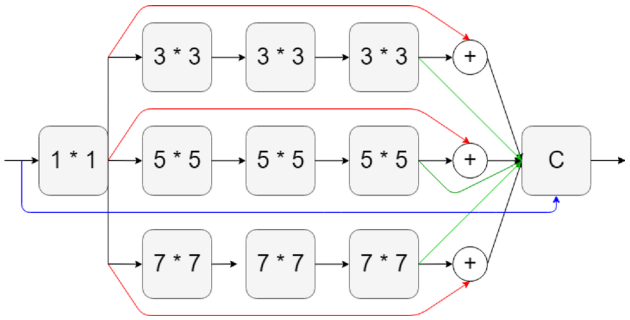

Fig. 1 A single module of ChoiceNet. The '+' denotes skip connections and 'C' denotes channelwise concatenation

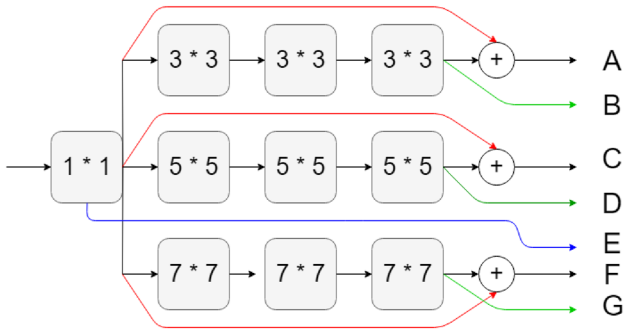

Fig. 2 A breakdown of the ChoiceNet module of Fig. 1. Here, letters A to G denote unique information generated by one forward pass through the module

2 Related works

Since the discovery of convolutional network, finding the ideal network architecture for a particular task has been a challenging area of research. The increased number of layers in modern architectures signifies the differences between different patterns of connectivity and revising older ideas.

ResNet ResNet [16] uses identity mapping as bypassing paths to improve over a typical CNN network [27].

A typical convolutional feed-forward network connects the lth layer's output to the $(l+1)$th layer's input. It gives rise to a layer transition: $x_l = H_l(x_{l-1})$. ResNet [16] adds an identity mapped connection, also referred as skip connection, that bypasses the transformation in between:

$$x_l = H_l(x_{l-1}) + x_{l-1}. \tag{1}$$

This mechanism allows the network to flow gradients directly through the identity functions which results in faster training and better error propagation. However, in [18] it was argued that despite the benefits of using skip connections, there is a possibility that when a layer is connected by a skip connection it may disrupt the information flow of the network therefore degrading the performance of the network.

In [56], a wider version of ResNet was proposed where the authors showed that an increased number of filters in each layer could improve the overall performance with sufficient depth. FractalNet [29] also shows comparable improvement on similar benchmark datasets[27].

DenseNet As an alternative to ResNet, DenseNet proposed a different connectivity scheme. They allowed connections from each layer to all of its subsequent layers. Thus *lth* layer receives feature maps from all previous layers. Considering $x_0, x_1 \ldots, x_{l-1}$ as input:

$$x_l = H_l([x_0, x_1 \ldots, x_{l-1}]) \tag{2}$$

where $[x_0, x_1 \ldots, x_{l-1}]$ denotes the concatenation of feature maps produced from previous layers, respectively.

The network maximises information flow by connecting the convolutional layers channelwise instead of skipping connections. In this model, the layer l has l number of inputs consisting of all the feature maps of previous $l-1$ layers. Thus on the *lth* layer, there are $l(l+1)/2$ connections. DenseNet requires fewer parameters as there is no need to learn from redundant features maps. This allows the network to compete against ResNet using fewer parameters.

We propose an alternative connectivity that retains the advantages of the above architectures while reducing some of their limitations. Figure 1 illustrates the connectivity layout between each layer of a single module. Each block of ChoiceNet contains three modules and the total network is comprised of three blocks with pooling operations in the middle (see Fig. 3).

3 ChoiceNet

Consider a single image x_0 that is passing through a CNN. The network has L layers, each with a nonlinear transformation $H_l(.)$, where l is the index number of the layer. $H_l(.)$ is a list of operations such as batch normalisation [20], Pooling [31], rectified linear units [39] or a convolutional operation. The output of lth layer is denoted as x_l.

ChoiceNet We propose an alternative connectivity that retains the advantages of the above architectures while reducing some of their limitations. Figure 1 illustrates the connectivity layout between each layer of a single module. Each block of ChoiceNet contains three modules and the total network is comprised of three blocks with pooling operations in the middle (see Fig. 3).

Figure 2 shows a breakdown of each module. Letters A to G denote unique information generated by one forward pass through the model. B is generated by three consecutive 3×3 convolutional operations, whereas A is the result of the same three convolutional operations but additionally connected by a skip connection. Following this pattern, we generate

information represented by letters C, D, F and G. Letter E denotes the special case where no convolutional operation is done after the 1×1 convolutional operation and it contains all the original information. This information is then concatenated with the others (i.e. A, B, etc.) at the final output.

Therefore, the final output contains information with and without skip connections from filters of size 3, 5 and 7 and also from the original input without any modification. Note that the 1×1 convolutional operation at the start acts as a bottleneck to limit computational costs and all the convolutional operations are padded appropriately for the concatenation at the final stage. Kernel sizes of 3, 5 and 7 were chosen because these three sizes together give the best performance [54, 55]. Adding more kernel sizes such as a combination of 3, 5, 9 and 11 or 3, 7 and 11 increases the network size in parameters without much improvement in performance (Fig. 4).

Considering $x_0, x_1 \ldots, x_{l-1}$ as input, our proposed connectivity is given by:

$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{3}$$

$$x_{l+1} = H_l([x_l, x_{l-1}]) + x_l \tag{4}$$

where $[x_l, x_{l-1}]$ is concatenation of feature maps. The feature maps are first summed and then concatenated which resembles characteristics of ResNet and DenseNet, respectively.

Composite function Each of the composite functions consists of a convolution operation followed by a batch normalisation and ends with a rectified linear unit(ReLU) operation.

Pooling Pooling is an essential part of convolutional networks since Eqs. (1) and (2) are not viable when the feature maps are not of equal size. We divide the network into multiple blocks where each block contains same sized features. Instead of using either max pooling or average pooling, we use both pooling mechanisms and concatenate them before feeding it to the next layer (see Fig. 5).

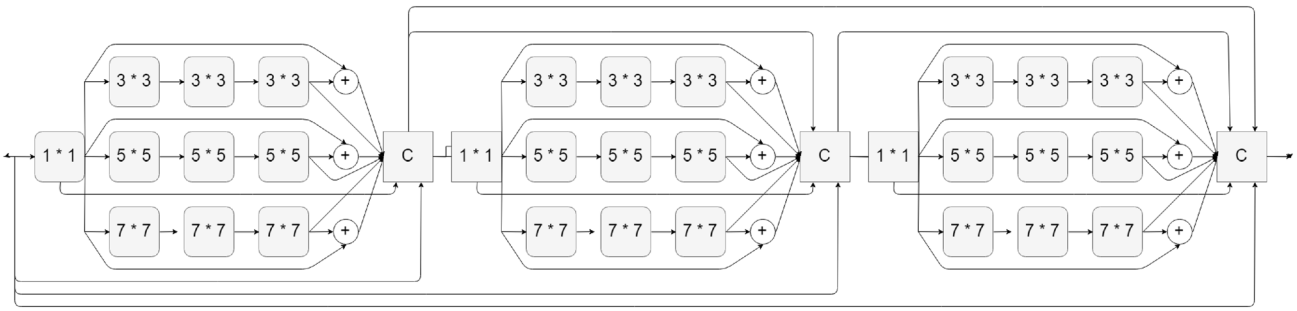

Fig. 3 A single block of ChoiceNet containing three consecutive ChoiceNet modules (see Fig. 1). They are simply stacked one after another and densely connected like DenseNet [18]

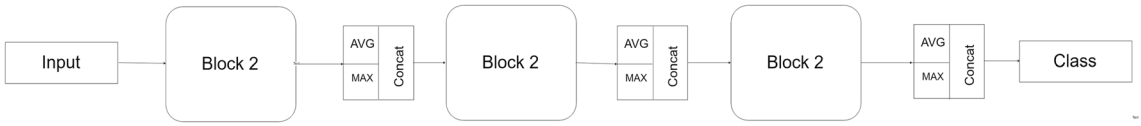

Fig. 4 The ChoiceNet consists of three ChoiceNet blocks where each block contains three ChoiceNet modules (see Fig. 3) and each ChoiceNet module is connected via feature maps and skip connections (see Fig. 1). After each block, there is a Max-pool and an Avg-Pool operation and their feature maps are concatenated for the next layer

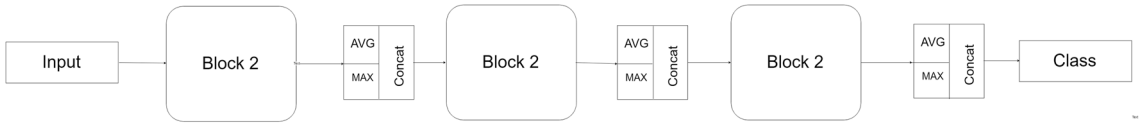

Fig. 5 The ChoiceNet consists of three ChoiceNet blocks where each block contains three ChoiceNet modules (see Fig. 3) and each ChoiceNet module is connected via feature maps and skip connections (see Fig. 1). After each block, there is a Max-pool and an Avg-Pool operation and their feature maps are concatenated for the next layer

Bottleneck layers The use of 1×1 convolutional operations (known as bottleneck layers) can reduce computational complexity without hurting the overall performance of a network [34]. We introduce a 1×1 convolutional operation at the start of each composite function (see Figs. 1 and 3).

Implementation details ChoiceNet has three blocks with equal number of modules inside. In each Choice operation (see Fig. 1), there are three 3×3, three 5×5 and three 7×7 convolutional operations. Each of the consecutive convolutional operations is connected via a skip connection (red line in Fig. 1). The feature maps are then concatenated so that both the outputs with the skip and without the skip connections are included (green and black lines in Fig. 1 before 'C'). Finally, the original input feature map is also merged (blue line in Fig. 1) to produce the final output.

The idea behind having the skip (Letter A, Fig. 2) and the non-skip connections (Letter B, Fig. 2) output merged together is for enabling the network to choose between the two options for each filter size. We also merge the original input to this output (Letter E, Fig. 2) so that the network can choose a suitable depth for optimal performance. To allow the network further options, we use both Max and average pooling. Thus, each pooling layer contains both a Max-Pool and an Avg-Pool operation. The outputs of each pooling operation are merged before proceeding to the next layer.

4 Experiments

We evaluate our proposed ChoiceNet architecture on benchmark datasets (ImageNet [28], CIFAR10 [27], CIFAR 100 [27] and SVHN [40]) and compared it with other state-of-the-art architectures. We also evaluated it on state-of-the-art semantic segmentation dataset CamVid [22] and 300W [47] dataset for facial landmark localisation.

4.1 Datasets

4.1.1 CIFAR

The CIFAR dataset [27] is a collection of two datasets, CIFAR10 and CIFAR100. Each dataset consists of 50,000 training images and 10,000 test images with 32×32 pixels. The CIFAR10 dataset contains 10 class values and CIFAR100 dataset contains 100. In our experiment, we hold out 5000 images from the training set for validation and use the rest of the images for training. We choose the model with the highest accuracy on the validation set to test on the test set. We adopt standard data augmentation with training including horizontally flipping images, random cropping, shifting and normalising using channel mean and standard deviation. These augmentations were widely used in previous work [16, 19, 29, 32, 34, 43, 51, 52]. We also tested our model on the datasets without augmentation. In our final output in Table 1, we denote the original dataset as C10 and C100, and the augmented dataset as C10+ and C100+.

4.1.2 SVHN

The SVHN dataset contains images of **S**treet **V**iew **H**ouse **N**umbers with 32×32 pixels. There are 73,257 images in the training set and 26,032 on the test set. It also contains additional 531,131 images for training purposes. Like in previous work [16, 19, 29, 32, 43], we use all the training data with no augmentation and use 10% of the training images as a validation set. We select the model with the highest accuracy on the validation set and report the test error in Table 1.

4.1.3 CamVid

The CamVid dataset [12] is a dataset consisting of 12 classes and has been mostly used for the task of semantic segmentation in previous work [2, 10, 38]. The dataset contains a training set of 367 images, a validation set of 100 images and a test set of 233 images. The challenge is to do pixelwise classification of the input image and correctly identify the objects in the scene. The metric called IoU or 'intersection over union' is commonly used for this particular task [2, 7, 22].

4.1.4 ImageNet

The ILSVRC 2012 classification dataset [46] consists of 1.2 million images for training and 50,000 for validation with 1000 classes. We adopt the same data augmentation scheme for training images as in [18, 19] and apply a single-crop or 10-crop with size 224×224 at test time. Following [18], we report classification errors on the validation set.

4.1.5 300W

The 300W [47–49] dataset is a collection of multiple face datasets such as LFPW [3], HELEN [25], AFW [62] and XM2VTS [37]. This is a challenging dataset that has been widely used for benchmarking facial landmark localisation algorithms [41]. The images in the dataset contain faces and 68 local landmarks [8, 9] semi-automatically annotated [49].

4.2 Training

Each of the experiments was performed 5 times and during the training process we took the model with the best validation score and reported its performance on the test set.

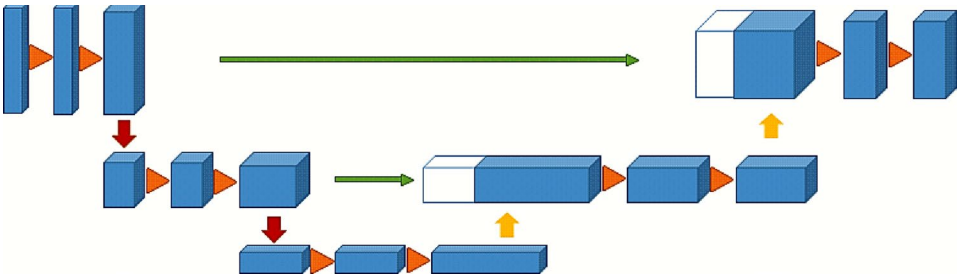

Fig. 6 Training procedure using U-Net [44]. Before each pooling operation, the features are stored and later concatenated when the feature maps are upsampled as indicated by the green arrows

Table 1 Error rates (100 - accuracy)% on CIFAR and SVHN datasets

Method	Depth	Params	C10	C10+	C100	C100+	SVHN
Network in Network	–	–	10.41	8.81	35.68	–	2.35
All-CNN	–	–	9.08	7.25	–	33.71	–
Deeply Supervised Net	–	–	9.69	7.97	–	34.57	1.92
Highway Network	–	–	–	7.72	–	32.39	–
FractalNet	21	38.6M	10.18	5.22	35.34	23.3	2.01
With Dropout/Drop-path	21	38.6M	7.33	4.6	28.2	23.73	1.87
ResNet	110	1.7M	–	6.61*	–	–	–
ResNet (reported by [19])	110	1.7M	13.63	6.41	44.74	27.22	2.01
ResNet with Stochastic Depth	110	1.7M	11.66	5.23	37.8	24.58	1.75
	1202	10.2M	–	4.91	–	–	–
Wide ResNet [61]	16	11.0M	6.29	4.81	–	22.07	–
	28	36.5M	–	4.17	–	20.5	–
With Dropout	16	2.7M	–	4.2	–	–	1.63
ResNet (pre-activation)	164	1.7M	10.5*	5.83	35.78	24.34	–
	1001	10.2M	10.4*	4.59	32.89	22.75	–
DenseNet ($k = 12$)	40	1.0M	7.0	5.24*	27.55*	24.42*	1.79*
DenseNet ($k = 12$)	100	7.0M	**5.77***	4.1*	23.79*	20.24*	1.67*
DenseNet ($k = 24$)	100	27.2M	5.83*	**3.74***	**23.42***	**19.25***	**1.6***
DenseNet-BC ($k = 12$)	100	0.8M	6*	4.51*	24.60*	22.98*	1.76*
DenseNet-BC ($k = 24$)	250	15.3M	**5.16***	3.9*	**19.75***	**17.60**	**1.74**
DenseNet-BC ($k = 40$)	190	25.6M	–	**3.7***	–	17.88*	–
Inception v3	–	13M	6.7	6.2	24.75	23.5	1.8
Inception v4	–	18M	6.4	6.0	24.40	23.22	1.75
ChoiceNet-30	30	13M	5.9	4.2	22.80	20.5	1.8
ChoiceNet-37	37	19.2M	*4.0*	*3.9*	*18.91*	*17*	*1.6*
ChoiceNet-40	40	23.4M	*3.9*	*3.2*	*18.05*	*16.6*	*1.5*

k denotes network's growth rate for DenseNet. Results that surpass all competing methods are italics and the overall best results are bold. '+' indicates standard data augmentation (translation and/or mirroring). The notation '*' indicates models run by ourselves. All the results of ChoiceNet without data augmentation(C10, C100, SVHN) are obtained using Dropout. ChoiceNet achieves lower error rates while using fewer parameters than ResNet and DenseNet. Without data augmentation, ChoiceNet performs better by a significant margin

4.2.1 Classification

All networks were trained using stochastic gradient descent (SGD) [5]. We avoid using other optimisers such as Adam [23] and RMSProp [15] to keep the comparisons as fair and simple as possible. On all three datasets, we used a training batch of 128. For the first 100 epochs, we used a learning rate of 0.001, for the next 100 epochs 0.0001, and then a rate of 0.00001 for the final 300 epochs.

We used weight decay of 0.0005 and Nesterov [45] momentum without dampening. We use a dropout layer after each ChoiceNet block with dropout rate at 0.2.

We use the learning parameters from [19] which was later used by [18] in order that the training environment is the same for every network.

4.2.2 Segmentation

For this task, we use the training procedure of U-Net [44] (Figure in supplementary materials S6) and we change the conv-blocks of U-Net with Res-Block (a block of the network that holds off the unique properties), Dense-Block and ChoiceNet-Module (Fig. 3). We use the Adam Optimiser with an initial learning rate of 0.001 which was reduced by a factor of 10 after each 100 epochs until the network converged. A weight decay of 0.0005 and Nesterov [45] momentum without dampening was used. For fair comparison, we kept the number of channels of Res-block and Dense-block unchanged as in the original article [18, 19] (Fig. 6).

4.2.3 Facial landmark prediction

For evaluation, we followed the protocol used in [13, 42] where the final test set consists of 689 images and is divided into two category such as *common* and *challenging*. The common subset has 554 images and the challenging set has the rest. We used L1 loss as it is more appropriate for this task [13] and we also used *Wing* − *Loss* [13] which is a robust loss function designed for facial landmark prediction.

5 Result analysis

5.1 CIFAR and SVHN

Accuracy Table 1 shows that the ChoiceNet depth 40 achieves the highest accuracy on all three datasets. The error rate on C10+ and C100+ is 4.0% and 17.5%, respectively, which is lower than error rates achieved by other state-of-the-art models. Our results on the original C10 and C100 (without augmentation) datasets are 2% lower than Wide ResNet and 5% lower than pre-activated ResNet. Our model ChoiceNet ($d = 37$) performs comparably well to DenseNet-BC with $k = 24$ and $k = 40$, whereas ChoiceNet ($d = 40$) outperforms all other networks.

Parameter efficiency Table 1 shows that ChoiceNet needs fewer parameters to give similar or better performance compared to other state-of-the-art architectures. For instance, ChoiceNet with a depth of 30 has only 13 million parameters yet it performs comparably well to DenseNet-BC ($k = 24$) which has 15.3 million parameters. Our best results were achieved by ChoiceNet ($d = 40$) with 23.4 million parameters compared to DenseNet-BC ($k = 40$) with 25.6 m, DenseNet ($k = 24$) with 27.2 m and Wide ResNet with 36.5 m parameters.

Over-fitting Deep learning architectures can often be prone to over-fitting however as ChoiceNet requires a smaller number of parameters, it is less likely to overfit the training datasets. Its performance on the non-augmented datasets appears to support this claim.

Exploding gradient While training ChoiceNet we observed that it occasionally suffers from an exploding gradient problem. ResNet and DenseNet were both trained using stochastic gradient descend(SGD) and a learning rate of 0.1 that was later reduced to 0.01 and 0.001 after every 100 epochs. However, we had to start training our network using a learning rate of 0.001 because setting the rate any higher was causing gradients to explode. We also had to reduce the learning rate to 0.0001 and then to 0.00001 after each 50 epochs instead of 100 to prevent the problem from reoccurring (Table 2).

The problem of exploding gradients is easier to handle than that of vanishing gradients. We used a smaller learning rate at the start and L2 regularisers with dropout layers ($p = 0.5$) which addressed the problem.

5.2 CamVid

We tested ChoiceNet on the CamVid dataset and compared it with other state-of-the-art networks [7, 7, 14, 21, 24, 33, 36, 57–59]. Mean IoU (m_IoU) scores are shown in Table 4.

Our network performs better than other neural network architectures, it was able to outperform DenseNet and ResNet both in terms of m_IoU score as well as in terms of parameter efficiency. Our ChoiceNet with 13 million parameters was able to perform better than networks almost twice its size.

Table 2 Error rates of Top1 and Top 5% on ImageNet dataset

Method	Top-1 error%	Top-5 error%
Inception v3 [55]	22.9	5.9
Inception v4 [53]	21.5	**5.7**
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	5.71
DenseNet-($K = 12$)	23.82	6.85
DenseNet-($K = 24$)	22.58	6.34
DenseNet-($K = 40$)	22.15	6.12
ChoiceNet-30	*21.30*	5.8
ChoiceNet-37	*21.21*	**5.7**
ChoiceNet-40	*20.53*	*5.5*

Results that surpass all competing methods are italics and the overall best results are bold. ChoiceNet achieves lower error top 1% error rates with all three versions and lower top 5% error with ChoiceNet-40

5.3 300W

ChoiceNet was tested on 300W, a state-of-the-art facial landmark localisation dataset where the goal is to predict 68 landmarks on an individual's face. The dataset has two test sets called 'common' and 'challenging' where the 'common' test set is known to have instances which are easier to predict (examples in supplementary materials S4). The 'challenging' test set has more challenging cases where the faces are occluded or not clearly visible (supplement S4). We also found out that due to the semi-automatic nature of annotation in some cases in the 'challenging' test set the ground truth is not very precise but our model predicted more accurately. We hypothesise that on occasion where our model gives precise predictions to imprecise test annotation this may have increased the error of our model since ground truth did not match with the prediction (see Fig. 7).

In recent articles such as [13], it has been suggested that L1 loss is more appropriate for facial landmark localisation task than L2 loss. Loss function such as Wing-Loss has been also been developed specifically for this particular task. We used both L1 and wing loss and found out that ChoiceNet performs favourably compared to other state-of-the-art CNN architecture as well as architectures purposefully designed to this particular task such as CNN6/7 [13].

6 Discussion

Model compactness As a result of the use of different filter sizes with feature concatenation and skip connections at every stage, feature maps learned by any layer in a block can be accessed by all subsequent layers. This extensive feature reuse throughout the network leads to a compact model.

Feature reuse ChoiceNet uses different filter sizes with skip connections and channel concatenation in each module (see Fig. 1). The kernel size of 3, 5 and 7 were found optimal in [54, 55] compared to combinations such as 3, 5, 9 and 11 or 3, 7 and 11 because the other combinations make the network costly without much improvement in performance. In order to have a deeper and visual understanding of its operation, we took the weights of the first block (in ChoiceNet-30) and normalised them to the range [0, 1]. After normalising the weights we mapped them to two sections, weights under 0.4 as white and over 0.4 as coloured—see Fig. 8. We assumed that the weights less than 0.4 will have insignificant effect on the total performance. The figure shows that after the very first 1×1 convolution operation on the raw input, the conv operations with channel size 7 have more effect than size 3 and 5. In the second module, all the conv operations' weights were under 0.4 which suggests that the model used either the feature maps of the earlier output by concatenation (red line between filter 5 and 7 of the middle module) or it used the skip connection (red line above filter 3 with highlighted '+' sign). On one hand, this indicates that the skip connection or channel concatenation or both are working as they were suppose to but this also means that we still have many redundant parameters in the network. In the third module, it was found that filters 3 and 5 had weights over 0.4 which indicates that they possibly had some contribution in the network. We suspect that the selection of filter size 7 in the first module and 3 and 5 in the third module echoes the hypothesis of AlexNet [28] where they found bigger filter sizes work better at the beginning of the networks and smaller filters work better in the later stages. However, the chosen path inside the network is not the same for every dataset. The bottom figure (see Fig. 8) displays the network trained for C10 (without augmentation). The dissimilarities show that even though they were trained on different versions (with and without augmentation) of the same dataset, the augmentation indirectly made the inside of the network quite different. Since it is very difficult to predict how the network will respond to a dataset, we cannot preselect a path before training for optimal performance therefore our design provides more *choice* within the network so that it can find the optimal path by itself.

Ablation Study 1 In Table 7, we provide an ablation study on the ChoiceNet-30. We use C10 and C10+ dataset for this purpose. We disable different parts of the network path such as A, B (see Fig. 2) and compare the performance with the original model. The column 'Difference between ChoiceNet-30' shows the increase in errors when a certain path is disabled thus imping that the higher the error rate without that complement the higher impact it has on the total performance. The table shows that on both C10 and C10+, the connection 'E' had the highest impact but all the other paths also had impacts as well which confirms that every path within the networks improves the network's performance. The small difference also suggests that all the paths the contributing to the same level and no individual paths are dominating.

Ablation Study 2 Similarly, in Table 3, we show the effect of the usage of two types of pooling method with our architectural design. We find that for all three models the use of max pool gave advantage over avg pooling. ChoiceNet-40 achieved the lowest error rate among the pooling techniques individually however it was superseded by the same model when both pooling were used. This shows even though, in cases, avg pooling may not be as effective as maxpool, using them together leads to improved performance.

In Table 4, we show the ***M**ean **I**ntersection **o**ver **U**nion* (m_IoU) on the CamVid dataset of some of the current state-of-the-art models. We used the U-Net training scheme and changed the basic convolutional operations with ResBlocks, DenseBlocks and ChoiceNet-module (see Fig. 1). While our network has fewer parameters compared to ResBlock

Table 3 Error rates of Top1 and Top 5% on ImageNet dataset of ChoiceNet with only Maxpool, AvgPool and both of them together

Method	Pooling	Top-1 Error%	Top-5 Error%
ChoiceNet-30	Max	21.45	6.0
ChoiceNet-37	Max	21.32	**5.9**
ChoiceNet-40	Max	**21.02**	**5.9**
ChoiceNet-30	Avg	22.25	6.2
ChoiceNet-37	Avg	22.21	6.1
ChoiceNet-40	Avg	21.80	**5.9**
ChoiceNet-30	Both	21.30	*5.8*
ChoiceNet-37	Both	21.21	*5.7*
ChoiceNet-40	Both	*20.53*	*5.5*

Results that surpass all competing methods are italics and the overall best results are bold. ChoiceNet achieves lower error top 1% error rates with all three versions and lower top 5% error with ChoiceNet-40

Table 4 The mean IoU (m_IoU) of all the classes on the CamVid dataset (test set) where mean-IoU means the mean of IoUs of all the 12 classes

Method	m_IoU
ChoiceNet-block	**73.5**
Inception v3 [55]	71.9
Inception v4 [53]	71.5
Lin et al. [33]	73.6
ResNet-152	70.6
ResNet-50	70.1
Dense-BC ($k = 40$)	69.2
Dense-BC ($k = 24$)	68.9
Dense-BC ($k = 12$)	68.5
Lo et al. [35]	67.3
Yu et al. [60]	67.1
Kreo et al. [26]	66.3
Chen et al. [6]	63.1
Berman et al. [4]	63.1
Arnab et al. [1]	62.5

Table 5 A comparison in error between different network architectures on 300W dataset with L1 Loss

Method	Common	Challenging	Full
ChoiceNet-30	3	6.1	**4.55**
ChoiceNet-37	**2.9**	**5.88**	**4.39**
ChoiceNet-40	**2.8**	**5.75**	**4.27**
Inception v3 [55]	3.5	8.1	5.8
Inception v4 [53]	3.3	7.8	5.55
Res-Net 50	3.4	*6.01*	*4.7*
Res-Net 152	*3.2*	*5.98*	*4.59*
DenseNet BC ($k = 12$)	4.8	9.52	9.56
DenseNet BC ($k = 24$)	4.5	9.02	6.76
DenseNet BC ($k = 40$)	3.9	8.75	6.32
CNN-6/7 (CVPR18) [13]	*3.2*	7.1	5.15

Results that surpass all competing methods are italics and the overall best results are bold. The column 'full' is the average of 'Common' and 'Challenging'

Table 6 A comparison in error between different network architectures on 300W dataset with *Wing* Loss

Method	Common	Challenging	Full
ChoiceNet-30	**3.4**	7.3	**5.35**
ChoiceNet-37	**3.2**	**6.9**	**5.05**
ChoiceNet-40	**3.2**	**6.7**	**4.95**
Inception v3 [55]	5	8.22	6.61
Inception v4 [53]	4.4	7.8	6.1
Res-Net 50	4.6	7.92	6.26
Res-Net 152	*3.5*	*7.26*	*5.38*
DenseNet BC ($k = 12$)	6.1	10.49	8.29
DenseNet BC ($k = 24$)	5.8	10.02	7.91
DenseNet BC ($k = 40$)	5.3	9.83	7.56
CNN-6/7 (CVPR18) [13]	*3.5*	*7.02*	*5.2*

Results that surpass all competing methods are italics and the overall best results are bold. The column full is the average of 'Common' and 'Challenging'

and Denseblocks, it achieved a higher score. Note that even though our model achieved a good m_IoU score, it is not as good as some of the network architectures designed specifically for segmentation tasks [21, 24, 57–59]. Nevertheless, it performed well comparing to both ResBlock and Denseblock as well as some other general purpose convolutional neural networks [36]. Some outputs are displayed in 'S1' section of supplementary materials.

In Tables 5 and 6, we show the performance of different state-of-the-art neural networks on 300W dataset using L1 and Wing-Loss, respectively. We also include methods such as CNN 6/7 which is specifically designed for this purpose with its robust loss function (Wing-Loss). The tables show that, with both loss function our model performs the highest on the 'full' dataset. ChoiceNet also achieves the lowest error on the 'Challenging' test set which further demonstrates the superiority of the proposed architecture. Detailed table and graph are displayed in 'S2' and 'S3' section of supplementary materials. In Fig. 7, we also show that in some cases the network predicts more precisely that the ground truth which increases the error rate as it doesn't match with the less precise ground truth (Table 7).

Our intuition is that the extra connections and paths in our method enable the network to learn from a large variety of feature maps. This also enables the network to backpropagate errors more efficiently (see also [16, 18]). We found that due to all the connections the network can be prone to exploding gradient and therefore needs a small learning rate to begin with. We also found by grid search that the network shows peak performance when the depth is between 30 and 40 layers and further increasing the layers appears to have

Fig. 7 A comparison between the ground truth of the 'challenging' test set (left) and ChoiceNet-40's prediction (right) where it shows that our model's prediction is sometimes more accurate than the semi-automatically annotated ground truth(GT) images but it also increases the error bar since it does not match with the GT

little effect. We suspect that ChoiceNet plateaus at depth 30–40 although it is possible that it could be a local minima as we couldn't train models with depth more than 60 layers due to resource limitation.

The performance on ImageNet dataset is displayed in Table 2. Our model with all three variation achieves lower top 1% score compared to other state-of-the-art neural network architectures like ResNet, DenseNet, Inception (v3/v4) and ChoiceNet-40 scores the lowest top 5% and top 1% error. This is a result of the unique connectivity design (see Fig. 2). Due to the usage of convolutional output with and without skip connection, using different kernel sizes, concatenating the original input per module via the connection 'E' of Fig. 2 and using two different pooling techniques together, it achieves this superior performance. Also as the architecture has many connections, therefore it can work with less channel outputs per convolution operation which makes it parameter efficient. This means given a number of parameters it achieves better performance than other methods.

7 Conclusion

In this paper, we introduced a powerful yet lightweight and efficient network, ChoiceNet, which encodes better spatial information from images by learning from its numerous elements such as skip connections, the use of different filter sizes, dense connectivity and including both Max and Avg pooling. ChoiceNet is a general purpose network with good generalisation abilities and can be used across a wide range of tasks including, but not limited to, classification, image segmentation, facial landmark localisation. Our network

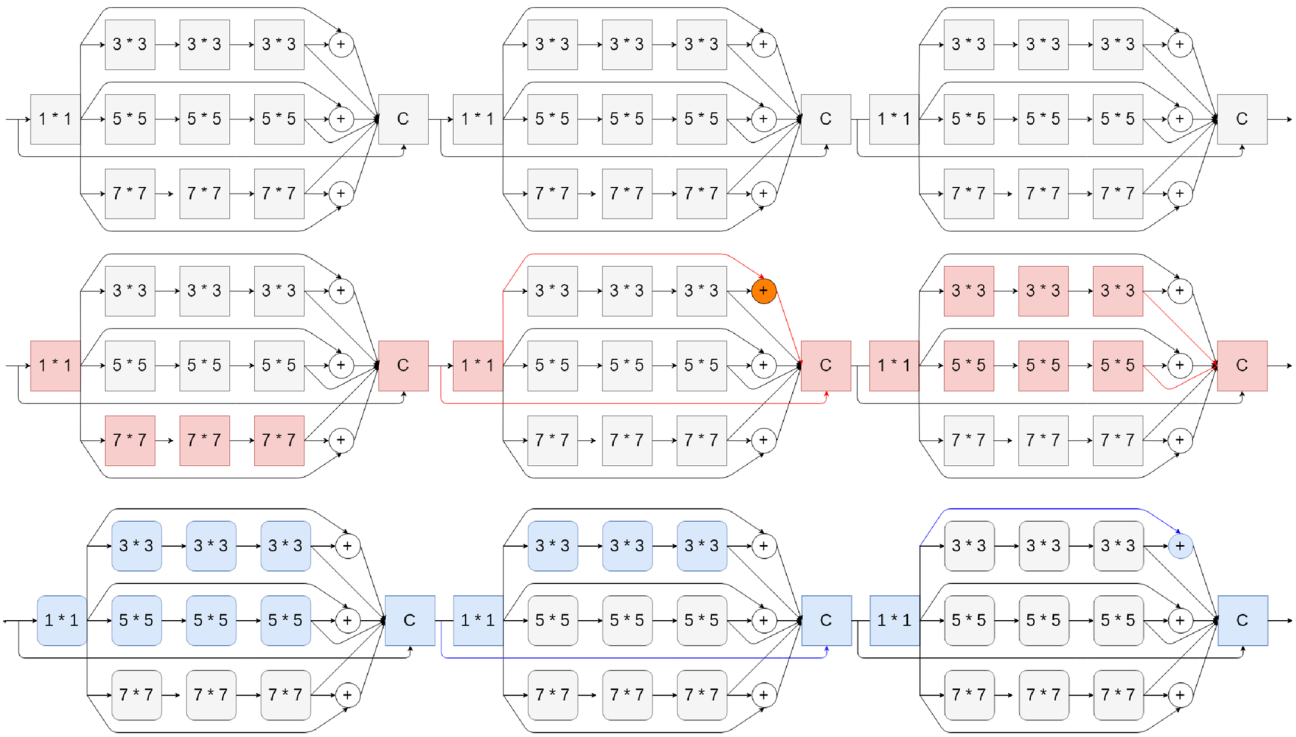

Fig. 8 An inside look at ChoiceNet. The top figure shows the skeleton of ChoiceNet before training and the bottom figure shows a path way the model has chosen for C10* (middle) and C10 (bottom) dataset for best classification accuracy after training. The coloured boxes and lines are putting the most contribution

Table 7 An error rate ablation study on ChoiceNet-30 on C10 and C10+

Method	C10	Diff between ChoiceNet-30	C10+	Diff between ChoiceNet-30
Without A	6.3	0.4	4.6	0.4
Without B	6	0.1	4.7	0.5
Without C	6.1	0.2	4.5	0.3
Without D	6.2	0.3	4.6	0.4
Without E	7.1	1.2	4.9	0.7
Without F	6	0.1	4.3	0.1
Without G	6.1	0.2	4.3	0.1
Full	5.9	0	4.2	0

The true performance of the network is denoted as 'full' and the column 'Difference between ChoiceNet-30' shows the difference in performance between the original performance and the same network without a certain connection path such as A/B

shows promising performance when compared to state-of-the-art techniques across different tasks such as semantic segmentation and object classification while being more efficient.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10044-021-01004-9.

References

1. Arnab A, Jayasumana S, Zheng S, Torr PHS (2016) Higher order conditional random fields in deep neural networks. In: European conference on computer vision (ECCV)
2. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495
3. Belhumeur PN, Jacobs DW, Kriegman DJ, Kumar N (2013) Localizing parts of faces using a consensus of exemplars. IEEE Trans Pattern Anal Mach Intell 35(12):2930–2940
4. Berman M, Rannen Triki A, Blaschko MB (2018) The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4413–4421
5. Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp 177–186
6. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2015) Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR
7. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
8. Cootes TF, Taylor CJ (1992) Active shape models-'smart snakes'. In: BMVC92. Springer, pp 266–275
9. Cootes TF, Taylor CJ, Cooper DH, Graham J (1995) Active shape models-their training and application. Comput Vis Image Underst 61(1):38–59
10. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3213–3223
11. Dahl GE, Sainath TN, Hinton GE (2013) Improving deep neural networks for lvcsr using rectified linear units and dropout. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8609–8613
12. Fauqueur J, Brostow G, Cipolla R(2007) Assisted video object labeling by joint tracking of regions and keypoints. In: 2007 IEEE 11th International conference on computer vision. IEEE, pp 1–7
13. Feng Z-H, Kittler J, Awais M, Huber P, Wu X-J (2018) Wing loss for robust facial landmark localisation with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2235–2245
14. Fourure D, Emonet R, Fromont E, Muselet D, Tremeau A, Wolf C (2017) Residual conv-deconv grid network for semantic segmentation. arXiv:1707.07958
15. Graves A (2013) Generating sequences with recurrent neural networks. arXiv:1308.0850
16. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
17. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
18. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: CVPR, vol 1, p 3
19. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ (2016) Deep networks with stochastic depth. In: European conference on computer vision. Springer, pp 646–661
20. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167
21. Ke T-W, Hwang J-J, Liu Z, Yu SX (2018) Adaptive affinity fields for semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp 587–602
22. Kendall A, Badrinarayanan V, Cipolla R(2015) Bayesian segnet: model uncertainty in deep convolutional encoder–decoder architectures for scene understanding. arXiv:1511.02680
23. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
24. Kong S, Fowlkes CC (2018) Recurrent scene parsing with perspective understanding in the loop. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 956–965
25. Koppen P, Feng Z-H, Kittler J, Awais M, Christmas W, Wu X-J, Yin H-F (2018) Gaussian mixture 3d morphable face model. Pattern Recogn 74:617–628
26. Krešo I, Čaušević D, Krapac J, Šegvić S(2016) Convolutional scale invariance for semantic segmentation. In: German conference on pattern recognition. Springer

27. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Technical report, Citeseer
28. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Sys 25:1097–1105
29. Larsson G, Maire M, Shakhnarovich G (2016) Fractalnet: ultra-deep neural networks without residuals. arXiv:1605.07648
30. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551
31. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
32. Lee C-Y, Xie S, Gallagher P, Zhang Z, Tu Z (2015) In: Proceedings of the eighteenth international conference on artificial intelligence and statistics, PMLR, vol 38, pp 562–570
33. Lin G, Milan A, Shen C, Reid I(2017) Refinenet: multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1925–1934
34. Lin M, Chen Q, Yan S (2013) Network in network. arXiv:1312.4400
35. Lo S-Y, Hang H-M, Chan S-W, Lin J-J (2018) Efficient dense modules of asymmetric convolution for real-time semantic segmentation. arXiv:1809.06323
36. Mehta S, Rastegari M, Shapiro L, Hajishirzi H (2018) Espnetv2: a light-weight, power efficient, and general purpose convolutional neural network. arXiv:1811.11431
37. Messer K, Matas J, Kittler J, Luettin J, Maitre G (1999) Xm2vtsdb: the extended m2vts database. In: Second international conference on audio and video-based biometric person authentication, vol 964, pp 965–966
38. Mulalić E, Grujić N, Ilić V, Marković M et al (2018) Object-level grouping and identification for tracking objects in a video, Feb. 20. US Patent 9,898,677
39. Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 807–814
40. Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning
41. Rayhan F, Galata A, Cootes TF (2020) Not all points are created equal-an anisotropic cost function for facial feature landmark location. In: The 31st British machine vision virtual conference. BMVC
42. Ren S, Cao X, Wei Y, Sun J (2016) Face alignment via regressing local binary features. IEEE Trans Image Process 25(3):1233–1245
43. Romero A, Ballas N, Kahou S. E, Chassang A, Gatta C, Bengio Y (2014) Fitnets: hints for thin deep nets. arXiv:1412.6550
44. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241
45. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv:1609.04747
46. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
47. Sagonas C, Antonakos E, Tzimiropoulos G, Zafeiriou S, Pantic M (2016) 300 faces in-the-wild challenge: database and results. Image Vis Comput 47:3–18
48. Sagonas C, Tzimiropoulos G, Zafeiriou S, Pantic M (2013) 300 faces in-the-wild challenge: the first facial landmark localization challenge. In: Proceedings of the IEEE international conference on computer vision workshops, pp 397–403
49. Sagonas C, Tzimiropoulos G, Zafeiriou S, Pantic M (2013) A semi-automatic methodology for facial landmark annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 896–903
50. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
51. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2014) Striving for simplicity: the all convolutional net. arXiv:1412.6806
52. Srivastava RK, Greff K, Schmidhuber J (2015) Training very deep networks. In: Advances in neural information processing systems, pp 2377–2385
53. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI, vol 4, p 12
54. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
55. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
56. Targ S, Almeida D, Lyman K (2016) Resnet in resnet: generalizing residual architectures. arXiv:1603.08029
57. Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, Cottrell G (2017) Understanding convolution for semantic segmentation. arXiv:1702.08502
58. Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, Cottrell G (2018) Understanding convolution for semantic segmentation. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 1451–1460
59. Wu Z, Shen C, Van Den Hengel A (2019) Wider or deeper: revisiting the resnet model for visual recognition. Pattern Recog 90:119–133
60. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122
61. Zagoruyko S, Komodakis N (2016) Wide residual networks. arXiv:1605.07146
62. Zhu X, Ramanan D (2012) Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 2879–2886